Parallel Programming with MatlabMPI *

Jeremy Kepner (kepner@ll.mit.edu) MIT Lincoln Laboratory, Lexington, MA 02420

July 23, 2001

Abstract

MatlabMPI is a Matlab implementation of the Message Passing Interface (MPI) standard and allows any Matlab program to exploit multiple processors. MatlabMPI currently implements the basic six functions that are the core of the MPI point-to-point communications standard. The key technical innovation of MatlabMPI is that it implements the widely used MPI "look and feel" on top of standard Matlab file I/O, resulting in an extremely compact (∼100 lines) and "pure" implementation which runs anywhere Matlab runs. The performance has been tested on both shared and distributed memory parallel computers. MatlabMPI can match the bandwidth of C based MPI at large message sizes. A test image filtering application using MatlabMPI achieved a speedup of ∼70 on a parallel computer.

1 Introduction

Matlab [1] is the dominant programming language for implementing numerical computations and is widely used for algorithm development, simulation, data reduction, testing and system evaluation. Many of these computations can benefit from faster execution on a parallel computer. There have been many previous attempts to provide an efficient mechanism for running Matlab programs on parallel computers [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. These efforts of have faced numerous challenges and none have received widespread acceptance.

In the world of parallel computing the Message Passing Interface (MPI) [2] is the de facto standard for implementing programs on multiple processors. MPI defines C language functions for doing point-to-point communication in a parallel program. MPI has proven to be an effective model for implementing parallel programs and is used by many of the worlds' most demanding applications (weather modeling, weapons simulation, aircraft design, and signal processing simulation).

MatlabMPI consists of a set of Matlab scripts that implements a subset of MPI and allows any Matlab program to be run on a parallel computer. The key innovation of MatlabMPI is that it implements the widely used MPI "look and feel" on top of standard Matlab file I/O, resulting in a "pure" Matlab implementation that is exceedingly small (∼100 lines of code). Thus, MatlabMPI will run on any combination of computers that Matlab supports.

2 System Requirements

On shared memory systems, MatlabMPI only requires a single Matlab license since any user is allowed to launch many Matlab sessions. On a distributed memory system, MatlabMPI requires one Matlab license per machine. Because MatlabMPI uses file I/O for communication, there must also be a directory that is visible to every machine (this is usually also required in order to install Matlab). This directory defaults to the directory that the program is launched from, but can be changed within the MatlabMPI program.

3 Performance Test Results

The vast majority of potential Matlab applications are "embarrassingly" parallel and require minimal performance out of MatlabMPI. These applications exploit coarse grain parallelism and communicate rarely (if at all). Nevertheless, measuring performance is useful for determining which applications are most suitable for MatlabMPI.

*This work is sponsored by the High Performance Computing Modernization Office, under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the Department of Defense.

MatlabMPI has been run on several Unix platforms. It has been benchmarked and compared to the performance of C MPI on the SGI Origin2000. These results indicate that for large messages (∼1 MByte) MatlabMPI is able to match the performance of C MPI (see Figure 1). For smaller messages, MatlabMPI is dominated by its latency (∼35 milliseconds), which is significantly larger than C MPI.

To further test MatlabMPI a test application using MatlabMPI in a simple image filtering application was written. The application executed repeated 2D convolutions on a large image (1000 x 128,000 pixels ∼ 2 GBytes). This program demonstrates that the MPI standard is valid within the Matlab environment and allows parallel programs to be written quickly and easily. Furthermore, this application achieved excellent speedups (greater than 64 on 64 processors) and shows the classic super-linear speedup (due to better cache usage) that comes from breaking a very large memory problem into many smaller problems (see Figure 2).

4 Discussion and Future Work

MatlabMPI demonstrates that the standard approach to writing parallel programs in C and Fortran (i.e. using MPI) is also valid in Matlab. In addition, by using Matlab file I/O, it was possible to implement MatlabMPI entirely within the Matlab environment, making it instantly portable to all computers that Matlab runs on. Most potential parallel Matlab applications are trivially parallel and don't require high performance. Never-the-less, MatlabMPI can match C MPI performance on large messages.

The use of file I/O as a parallel communication mechanism is not new and is now increasingly feasible with the availability of low cost high speed disks. The extreme example of this approach are the now popular Storage Area Networks (SAN), which combine high speed routers and disks to provide server solutions. Although using file I/O increases the latency of messages it normally will not effect the bandwidth. Furthermore, the use of file I/O has several additional functional advantages which make it easy to implement large buffer sizes, recordable messages, multi-casting, and one-sided messaging. Finally, the MatlabMPI approach is readily applied to any language (e.g. IDL, Python, and Perl).

The simplicity and performance of MatlabMPI makes it a very reasonable choice for programmers that want to speed up their Matlab code on a parallel computer. Further work will aim at enhancing the performance of MatlabMPI and increase the number MPI functions.

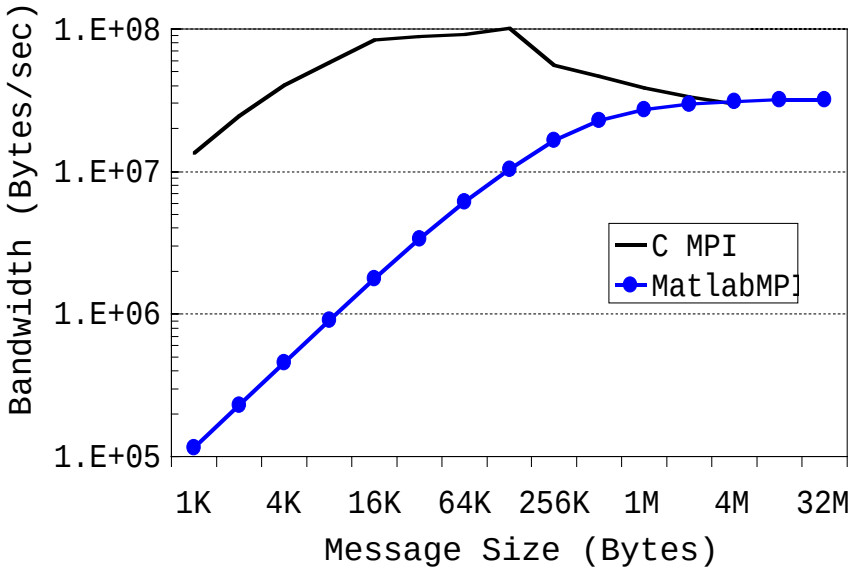

Figure 1: **Bandwidth.** Communication performance as a function message size on the SGI Origin2000. MatlabMPI equals C MPI performance at large message sizes.

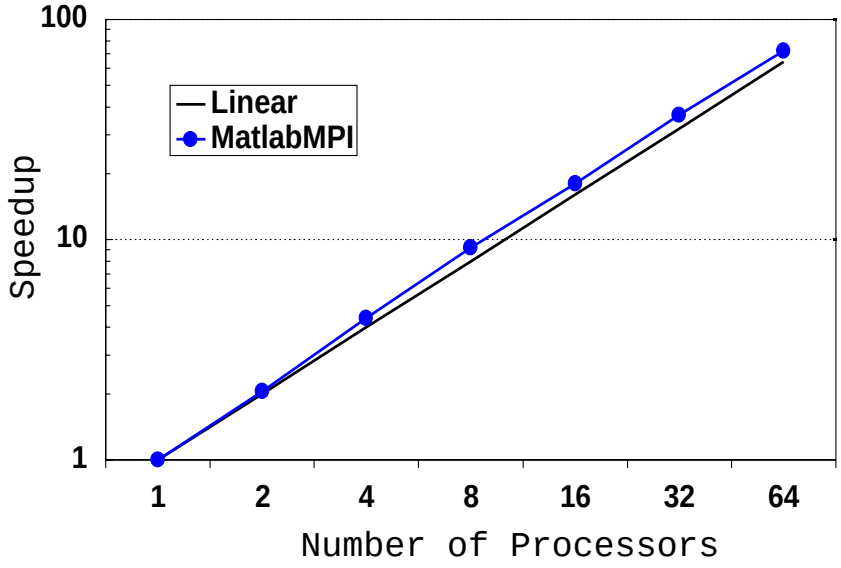

Figure 2: **Parallel Speedup.** Speed increase on the SGI Origin2000 of a parallel image filtering application as a function of the number of processors. Application shows "classic" super-linear performance (due to better cache usage) that results when a very large memory problem is broken into multiple small memory problems.

References

[1] Matlab, The MathWorks, Inc., http://www.mathworks.com/products/matlab/

[2] Message Passing Interface (MPI), http://www.mpi-forum.org/

[3] MATLAB*P, A. Edelman, MIT, http://www-math.mit.edu/∼edelman/

[4] A Parallel Linear Algebra Server for Matlab-like Environments, G. Morrow and Robert van de Geijn, 1998, Supercomputing 98 http://www.supercomp.org/sc98/TechPapers/sc98_FullAbstracts/Morrow779/index.htm

[5] Automatic Array Alignment in Parallel Matlab Scripts, I. Milosavljevic and M. Jabri, 1998

[6] Parallel MATLAB Development for High Performance Computing with RTExpress, http://www.rtexpress.com/

[7] MATLAB Parallel Example, Kadin Tseng, http://scv.bu.edu/SCV/Origin2000/matlab/MATLABexample.shtml

[8] MultiMATLAB: MATLAB on Multiple Processors A. Trefethen et al, http://www.cs.cornell.edu/Info/People/lnt/multimatlab.html

[9] ParaMat, http://www.alphadata.co.uk/dsheet/paramat.html

[10] Investigation of the Parallelization of AEW Simulations Written in MATLAB, Don Fabozzi 1999, HPEC99

[11] Matpar: Parallel Extensions to MATLAB, http://hpc.jpl.nasa.gov/PS/MATPAR/

[12] MPI Toolbox for Matlab (MPITB), http://atc.ugr.es/javier-bin/mpitb_eng

[13] A MATLAB Compiler for Parallel Computers. M. Quinn, http://www.cs.orst.edu/∼quinn/matlab.html

[14] Cornell Multitask Toolbox for MATLAB (CMTM), http://gremlin.tc.cornell.edu/er/media/2000/cmtm.html